

УДК 004.925

## Особливості архітектурної побудови систем формування тривимірних зображень

Романюк О.Н., Обідник М.Д., Романюк О.В.

Вінницький національний технічний університет

Костюкова Н.С.

Донецький національний технічний національний університет

### Abstract

*Romanyuk O.N., Obidnik M.D., Romanyuk O.V., Kostyukova N.S. Architecture features of three-dimensional image generation systems. In work the features of functional-architectural construction of three-dimensional picture generation systems, basic approaches to synthesis of graphic stages paralleling, technologies of two and more graphics cards union for the joint creating of one image are considered. It is shown, that the rise of productivity and realistic can be attained due to the use of new approaches for the three-dimensional picture generation and rise of functional possibilities and productivity of shade processors. Expansion of functionality of shade processors will give possibility to realize more realistic images, decrease loading on central processing unit and tire of exchange by data between the system board and graphics card*

### Вступ

У сучасних умовах комп'ютерна графіка стрімко розвивається, оскільки забезпечує високий рівень інформативності, відтворює найвагоміші аспекти процесу або явища. Найбільшу реалістичність забезпечує тривимірна графіка, основана на роботі з просторовими моделями об'єктів, що обумовлює її широке використання в багатьох галузях, пов'язаних з комп'ютерним моделюванням та проектуванням.

Дуже стрімко розвивається такий напрямок комп'ютерного синтезу зображень, як тривимірна комп'ютерна графіка реального часу, досягнення якої використовують при побудові систем візуалізації тренажерів транспортних засобів, моделюванні різних процесів, виробництві реклами і кіно, розробці комп'ютерних ігор і ін.

Головною особливістю тривимірної комп'ютерної графіки є робота з моделями об'єктів, розміщених у тривимірному просторі. На етапі проектування сцени сучасні засоби не забезпечують формування підсумкового зображення, що викликано необхідністю врахування багатьох факторів та великою обчислювальною складністю. Тому формування підсумкового зображення здійснюється на етапі візуалізації.

Особливості побудови систем візуалізації полягають у виконанні комплексу вимог: висока частота генерації зображення, мала затримка часу від моменту надходження керуючого впливу до адекватної зміни зображення, висока реалістичність синтезованого зображення, можливість відображати сцену великого обсягу і детальності.

На даному етапі розвитку тривимірної графіки відеокарти розвиваються в основному

екстенсивним шляхом. За останні кілька років архітектура відеокарт не зазнавала принципових змін. Усі модифікації зводилися до вдосконалювання апаратних конвеєрів відеокарти. Серед напрямків підвищення продуктивності графічних ядер, які отримали найбільшого поширення за останні роки, можна виділити такі: збільшення тактової частоти ядра, насамперед, завдяки вдосконаленню технологічного процесу виготовлення мікросхем; збільшення внутрішнього паралелізму виконання операцій шляхом збільшення кількості шейдерних процесорних блоків; забезпечення зовнішнього паралелізму, тобто можливості одночасної роботи декількох графічних процесорів над одним кадром.

Перші два напрямки обмежені технологічними можливостями виробництва графічних чипів. Збільшення кількості шейдерних процесорів дозволяє підвищити продуктивність формування тривимірних зображень, але при цьому завжди мають місце технологічні обмеження та підвищене тепловиділення, проблему якого не вирішено. У 2008 році з ноутбуків вилучені чипи NVIDIA M84 і M86, які через перегрів вивели з ладу ноутбуки, а графічні чипи останнього покоління G92 і G94 вилучені через часті збої у функціонуванні також через перегрів.

Підвищення продуктивності шляхом реалізації зовнішнього паралелізму може бути використано тільки для окремих випадків, оскільки пов'язано зі збільшенням споживаної потужності, габаритних розмірів системи і її вартості. Таке рішення неприйнятне для мобільних пристроїв, ринок яких постійно зростає. У мобільних пристроях із міркувань геометричних розмірів, ціни, тепловиділення й енергоспоживання недоцільно ставити окрему графічну карту.

## Постановка задачі

Сучасні засоби формування тривимірних зображень не забезпечують в повній мірі вимоги практики. Особливо це стосується формування динамічних зображень у реальному масштабі часу.

Це обумовлює необхідність аналізу особливостей архітектурної побудови систем формування тривимірних зображень з метою виявлення її властивостей та пошуку напрямків удосконалення архітектур для поліпшення параметрів, у першу чергу, підвищення продуктивності.

## Виклад основного матеріалу

При програмній реалізації верхніх рівнів конвеєра апаратні рендери були завантажені частково, що змушувало розробників знижувати геометричну деталізацію 3D-об'єктів. У 2000 році фірма nVidia випустила відеокарту GeForce 256 [1], до якої вперше застосовано назву GPU, оскільки графічний процесор повністю підтримував усі стадії графічного конвеєра. Це була перша відеокарта з інтегрованими апаратними геометричним процесором і процесором освітлення (T&L), які реалізували відповідні етапи конвеєра.

У 2001 році в відеокарті GeForce 3 вперше було реалізовано технологію nFinite FX [2], яка давала можливість гнучкого керування геометричним процесором за рахунок використання піксельних і вершинних шейдерів [3, 4]. До появи відеочипів з апаратною підтримкою піксельних шейдерів, у розробників були лише можливості звичайного мультитекстурування й альфа-блендингу [5]. Шейдери мали чотири версії [4, 5], основні характеристики яких наведено в таблиці 1. З таблиці видно, що з кожною версією стандарту DirectX збільшувалося кількість інструкцій і

змінних піксельних і вершинних шейдерів, а також їх можливостей.

У 2006 році графічний конвеєр зазнав революційних змін [6]. У нього було введено новий етап – оброблення полігонів, який виконує геометричний шейдер. Геометричні шейдери дозволяють обробляти дані на рівні примітивів. Структуру нового конвеєра зображено на рис. 1.

До появи Direct 10 вершинні шейдери не мали інформації про те, до яких трикутників належить дана вершина і які вершини є її сусідами. Крім того, вершинний шейдер не міг видалити вершину або додати нову, він міг тільки змінювати її позицію й інші атрибути.

Згідно з Direct 10 вершинний шейдер передає результат геометричному шейдеру, який дозволяє змінити топологію об'єкта, тобто створити нові примітиви та видалити існуючі. Став можливим доступ до сусідніх примітивів, тобто певний примітив, наприклад, трикутник, може зберігати інформацію про суміжні з ним трикутники.

У графічному конвеєрі на стадії компонування вхідних даних окрім взаємодії із центральним процесором з'явилася можливість отримання вершинних даних із буфера вершин або з буфера індексів.

Це дає можливість повторно вводити на конвеєр дані, розраховані вершинними, піксельними й геометричними шейдерами, завантажені за допомогою потокового виведення (на рис. 1 стадія введення даних в відеопам'ять). Завдяки цьому можна знову вводити на конвеєр ті самі дані, не повторюючи їх розрахунок, що, звичайно ж, розвантажує сам конвеєр і збільшує продуктивність. У випадку уніфікованої конвеєрної й шейдерної архітектури з'являється можливість зменшити кількість задіяних стадій конвеєра й

Таблиця 1. Характеристики версій шейдерів

Версія API	DirectX 8	DirectX 9	DirectX 9.0c	DirectX 10
Версія шейдерів	1.1/1.4	2.0	3.0	4.0
Вершинних інструкцій	128	256	512	65536
Піксельних інструкцій	4+8	32+64	512	65536
Вершинних констант	96	256	256	16x4096
Піксельних констант	8	32	224	16x4096
Вершинних змінних	16	16	16	4096
Піксельних змінних	2	12	32	4096
Потоків вершин	16	16	16	16
Потоків пікселів	4+2	8+2	10	32
Вершинних текстур	-	-	4	128
Піксельних текстур	8	16	16	128
Розмір текстури, точок			2048x2048	8192x8192
Керування обробкою вершин	-	статичне	статичне або динамічне	динамічне
Довільний доступ до даних із шейдерів	-	-	-	+

змінити лінійну послідовність обробки потоку даних.

Існує два обмеження в конвеєрній реалізації графічних систем. По-перше, базові алгоритми формування 3D-зображень легко розділити тільки на невелику кількість послідовних етапів, що дозволяє використати невелику кількість процесорів. По-друге, складові етапи мають різну трудомісткість обчислення, що породжує проблему збалансованого завантаження етапів конвеєра.

Коли запити досягають граничних можливостей одного конвеєра, то для прискорення формування зображень використовують декілька паралельно працюючих конвеєрів [6, 7, 8]. Ураховуючи надзвичайно великий обсяг обчислень при формуванні тривимірних графічних зображень, у сучасних графічних системах використовують паралельне оброблення інформації.

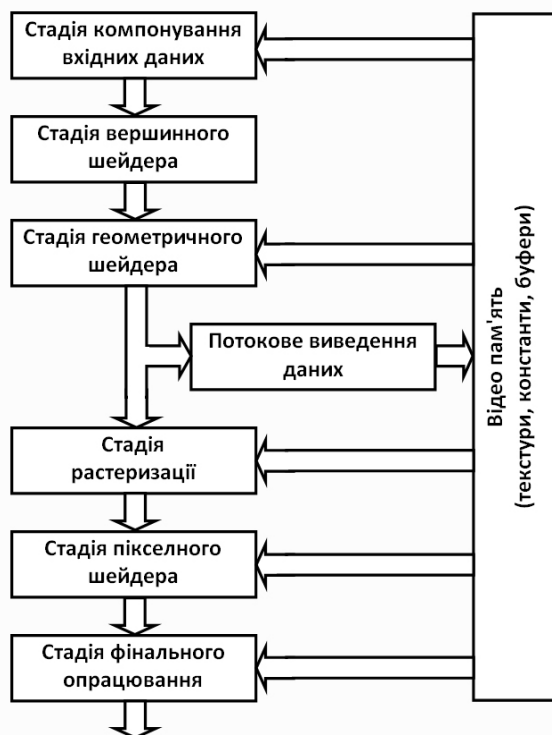


Рисунок 1 – Структура конвеєра

Можна виділити кілька основних підходів до розпаралелення: процедурний, який передбачає конвеєрну обробку інформації, коли кожний процесор реалізує функції окремого етапу формування сцени; композиційний, коли примітиви або їх сукупність розподіляються між процесорами; фрагментний, який передбачає розподіл екранного простору на фрагменти, кожний з яких формується своїм процесором, з подальшим об'єднання фрагментів в одну сцену.

Розрізняють такі основні рівні паралелізму [9, 10], які використовуються при розпаралеленні різних етапів формування реалістичних зображень: 1) рівень ЕЗ (піксела), коли виконується паралельне

обчислення по шести координатах (x, y, z, R, G, B); 2) рівень групи ЕЗ, де значення інтенсивностей і координат обчислюються паралельно для декількох ЕЗ;

3) рівень примітива, де паралелізм реалізується, наприклад, усередині трикутника, забезпечуючи паралельне обчислення для його горизонтальних і вертикальних лінійних або прямокутних фрагментів;

4) рівень групи примітивів, де паралелізм забезпечується одночасним растріванням ряду примітивів; 5) рівень кадру, де паралелізм досягається одночасним обробленням ряду підкадрів чи повних кадрів.

На рис. 2, а показано приклад розпаралелення з використанням множини конвеєрів [11], у яких геометричний процесор ГП виконує геометричні перетворення над трикутниками; на етапі растреризації Р визначаються адреси точок ділянки, обмеженої трикутником; вилучення невидимих поверхонь (ВНП) виконується з використанням Z-буфера, а видимі піксели запам'ятовуються в кадровому буфері КБ.

Можливо використання маршрутизації примітивів у відповідні растреризатори (з'єднання А на рис. 2, а), або маршрутизації фрагментів після растреризації у відповідні пристрої (з'єднання В на рис. 2, а).

На рис. 2, б зображено архітектуру графічної системи [12] фірми Silicon Graphics. Геометричні процесори виконують геометричні перетворення і передають дані генераторам фрагментів ГФ через шину трикутників. Буфер кадру поділено на тайли, кожний з яких керується своїм процесором.

До системи, у якій використано розпаралелення в просторі зображення, можна віднести систему Fuchs [13], у якій кожний з M.N процесорів зв'язується з кожним M-пікселом на кожному N рядку розгортки. Всі полігони сцени обробляються послідовно, причому багатокутник стає доступним відразу для всіх процесорів одночасно. Оскільки кожний полігон покриває неперервну частину екрана, то він буде оброблятися відразу усіма процесорами. Пам'ять при цьому організовується так, щоб кожний із чипів включав рівно по одному біту для будь-якої ділянки бітової площини тайлу. В такому випадку буде мати місце однозначна відповідність між чипами пам'яті й пікселями на екрані.

На рис. 2, с зображено архітектуру графічної системи Freedom [11], у якій об'єднано геометричні перетворення примітива та його растреризація. Декілька модулів працюють над декількома примітивами паралельно. Створені фрагменти маршрутизуються в модулі буфера кадру шляхом так званого поєднання буферів, що реалізовано в вигляді комутатора. Можливим конфліктом у роботі системи може стати передача від двох або більше растреризаторів даних до одного

модуля буфера кадру. Для цього випадку повинні бути передбачені засоби буферизації даних. Затрати, які необхідні для цього, зростають у нелінійній залежності, що обмежує масштабованість таких систем. За замовленням NASA для систем моделювання польотами було розроблено графічну систему [10], у якій 40 об'єктів було пов'язано з окремим процесором, кожний з яких являв собою достатньо складний пристрій, який формував динамічне зображення зі швидкістю відеогенерації.

Високою продуктивністю характеризуються графічні системи з використанням SLAM-чипів [10] – СВІС, які включають матрицю пам'яті з довільним доступом і вбудовані прості процесори, що працюють паралельно. Дисплейна система, буфер кадру якої складається з SLAM-чипів, дозволяє модифікувати одночасно від одного до кількох тисяч пікселів. Тонування багатокутників виконується на основі операцій зафарбовування рядка растрової розгортки з виконанням 4 SLAM-команд. Перша команда визначає рядок растрової

розгортки, який необхідно зафарбувати, друга – шаблон, який буде використовуватися для зафарбовування всіх модифікованих пікселів, третя – визначає кінцеву X-координату фрагмента рядка, який потрібно модифікувати, четверта – визначає початкову X-координату фрагмента рядка й пропонує SLAM виконати зафарбовування пікселів від початкової до кінцевої точок фрагмента шляхом послідовного повторення шаблону. У графічній системі [15], побудованій на основі SLAM-чипів, за один цикл пам'яті можна модифікувати довільну частину рядка.

Розглянемо архітектури відеокарт GeForce 7800 [8] і GeForce 8800 [6, 14], оскільки вони мають принципову різну організацію, характерну для різних поколінь відеокарт, причому відеокарти GeForce 8800 за підрахунками маркетингової компанії Jon Peddie Research у 2007-2008 роках мали найбільший обсяг продажу.

Вважається, що архітектура GeForce 8800 буде типовою й для наступного покоління відеокарт.

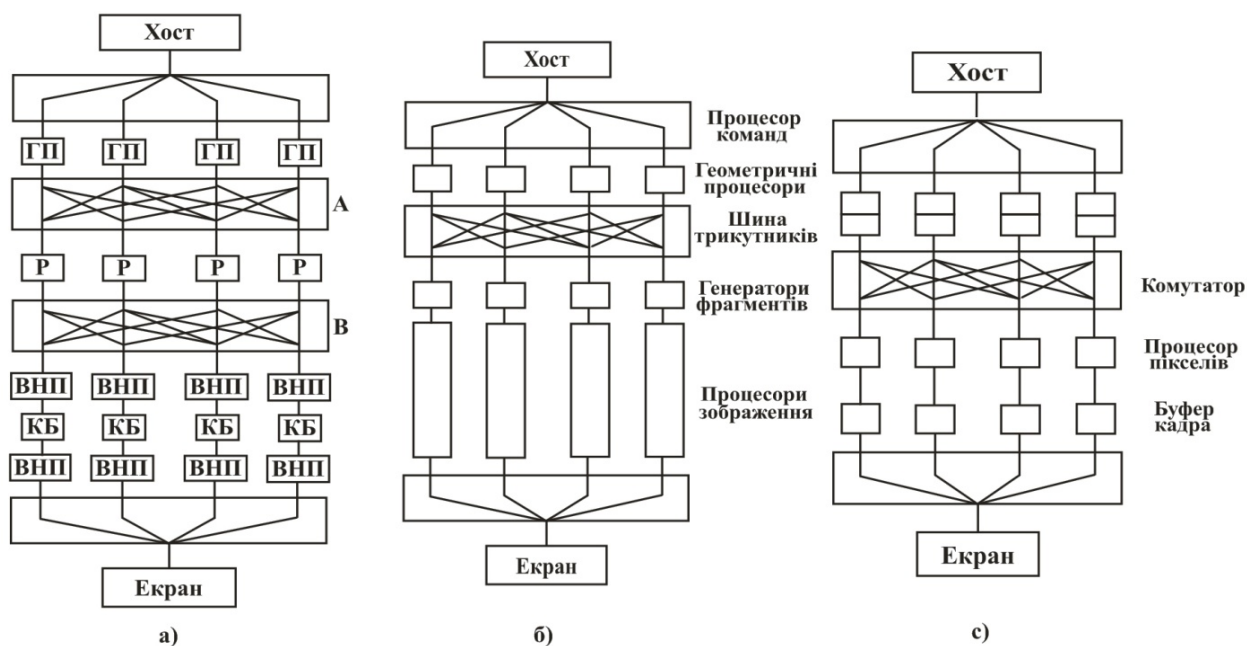


Рисунок 2 – Структури графічних систем

У відеокарті GeForce 7800 (2005 рік) [6] реалізовано графічний конвеєр із використанням вершинних і піксельних процесорів.

Відеокарта має 24 піксельних процесори PS, по одному текстурному блоці на конвеєр, 8 вершинних процесорів і 16 блоків растрових операцій (ROP). Піксельні процесори згруповано по 4 для обробки квадів [6]. Процесор PS має два векторні АЛП, здатні виконувати 2 різні операції над 4 компонентами та два міні-АЛП (найпростіші

скалярні АЛП для виконання простих операцій). Кожний піксельний блок може виконувати інструкції типу MADD [4, 6] (множення/додавання). АЛП 3 використовують для формування оптичних ефектів. АЛП 1 за один такт можна або вибрати одне значення текстури й задіяти другий АЛП 2 для однієї або двох операцій, або використати обидва АЛП, якщо не вибирається текстура.

За один такт вершинний процесор може виконати одну векторну операцію, одну скалярну

операцію й здійснити один доступ до текстури. До появи GeForce 8800 усім GPU було властиве одне фундаментальне обмеження [14] – розподіл виконавчих пристроїв для піксельних і вершинних шейдерів. Відповідно, будь-який графічний процесор містив у своєму складі два окремих набори блоків для обробки кожного виду шейдерів. Такий розподіл міг негативно позначатися на загальній ефективності роботи GPU, оскільки в сценах, насичених піксельними шейдерами, продуктивності наявних піксельних процесорів могло не вистачати, у той час як обчислювальні потужності вершинних процесорів не використовувалися, і навпаки.

Проблему дисбалансу вирішила уніфікація шейдерних процесорів, при якій навантаження між ними розподілялися динамічно, залежно від особливостей конкретної сцени. У відеокарті GeForce 8800 [6, 14] вперше використано уніфіковану шейдерну архітектуру рендерингу, потокове оброблення інформації та новий вид шейдера – геометричний. Чип складається з 8 універсальних процесорів, які включають 128 ALU і 32 TMU. Гранулярність виконання складає 8 блоків, кожний з яких може виконувати функції вершинного, піксельного, або геометричного шейдера над блоком із 32 пікселів.

Його називають шейдерним процесором. Кожний такий процесор має кеш першого рівня L1, у якому зберігаються текстури й дані, які можуть бути використані шейдерним процесором. Блоки ROP визначають факт видимості, запис у буфер кадру й мультисемплінг.

Вони згруповані з контролерами пам'яті, чергами запису та кешем другого рівня L2. Потокові процесори SP є уніфікованими скалярними процесорами із плаваючою комою, що обробляють не тільки графічні, але й інші дані. Об'єднання SP у кластери дозволяє ефективно використовувати апаратні ресурси відеокарти. Кожний потоковий процесор на основі механізмів керування здатний динамічно перепризначуватися для виконання конвексних графічних або інших операцій. Thread Processor керує завантаженням потокових процесорів.

Крім шейдерних блоків і ROP у GeForce 8800 [14] є набір керувальних блоків: Input Assembler приймає вихідні дані з пам'яті системи або локальної пам'яті; Setup/Raster/ZCull – блок, що виконує встановлення, растеризацію трикутника на блоки по 32 пікселі; блоки, що запускають на виконання програми даних різних форматів: вершинні (Vertex Thread Issue), геометричні (Geometry Thread Issue) і піксельні (Pixel Thread Issue).

Один із п'яти АЛП може виконувати такі спеціальні функції, як синус, логарифм, експонента й т.д. Інші чотири АЛП виконують операції додавання-множення. Кожний із суперскалярних

процесорів додатково має блок розгалужень, що підвищує ефективність роботи на шейдерах із великим числом переходів. Суперскалярна архітектура досягає найбільшої ефективності тоді, коли всі АЛП зайняті виконанням незалежних операцій, а домогтися цього досить складно, тому що в 3D-додатках багато операцій залежить від результатів виконання попередніх. Саме тому графічним ядрам ATI Radeon HD для досягнення найкращих результатів потрібна ретельна оптимізація драйверів під конкретний додаток.

На середину 2008 року найпотужнішою з дискретних вважається відеокарта Radeon HD 4870 X2, яка має продуктивність в 2,4 терафлопс. Компанія NVIDIA літом 2008 року офіційно представила новий інтегрований графічний процесор GeForce® 9400M, який забезпечує продуктивність в 54 GFLOP, що є абсолютним рекордом для мобільних рішень. Порівнюючи потужності флагманів відеокарт, можна стверджувати, що перша є в 44 рази потужнішою. Сьогодні рішення з вбудованими графічними ядрами по продуктивності досягають тільки рівня дискретних відеокарт нижнього класу.

На думку фахівців фірми Intel уже сьогодні необхідно, по крайній мірі, на порядок підвищити продуктивність інтегрованих графічних ядер [15]. Це важливо ще й тому, що в останні два роки загальні поставки інтегрованих графічних карт приблизно в три рази перевищили поставки дискретних графічних карт.

Наведені цифри свідчать, що сьогодні необхідно, щонайменше, в 23 рази підвищити продуктивність інтегрованих і мобільних графічних відеокарт для досягнення необхідного для користувача мінімального рівня динаміки графічних сцен при розподільній здатності 1024x768.

Потребують більш як удвічі підвищення продуктивності і дискретні відеокарти, які працюють у режимі високої якості.

Оскільки потужностей сучасних відеокарт недостатньо для високоякісної динамічної графіки, то іноді використовують технології Scan Line Interleaving (SLI) [16] і CrossFire [17]. Це технології об'єднання двох або більше відеокарт в одній системі для спільної роботи над одним зображенням. При цьому пропускна здатність шини ділиться між кількома відеокартами, кожна з яких працює за протоколом PCI Express. У 2007 році з'явився тандем трьох відеокарт – 3-Way SLI, а у 2008 року були анонсовані Triple і Quad CrossFire, які об'єднують відповідно три й чотири відеокарти Radeon. Технології SLI і CrossFire забезпечують підвищення продуктивності, але значно збільшують споживану потужність, габаритні розміри і не прийнятні для мобільних пристроїв.

На рис. 3 зображено структуру графічної системи, яку реалізовано за технологією CrossFire.

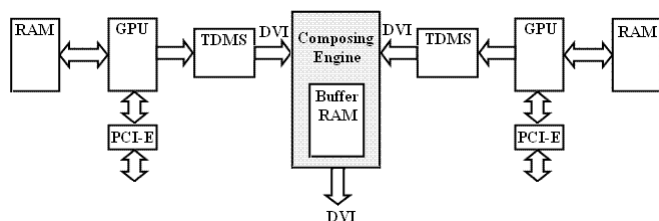


Рисунок 3 – Структура графічної системи, у якій використано технологію CrossFire

Кілька прискорювачів (на рис. 3 їх зображено два) формують власну частину зображення і виводять її через TDMS-трансмітери в цифровому стандарті DVI. Інформація подається в Composing Engine – пристрій, який здійснює об'єднання результатів роботи прискорювачів для одержання фінального зображення. На виході Composing Engine формується DVI-сигнал для фінального кадру, зібраного із двох порцій даних, розрахованих GPU. Для усунення проблем із синхронізацією Composing Engine містить власну буферну пам'ять, що дозволяє накопичувати дані асинхронно, а, потім, у міру готовності обох прискорювачів, формувати й видавати остаточний кадр. Composing Engine територіально розміщують на одній відеокарті, яку називають провідною. Дані із другої карти передають на провідну через звичайний зовнішній DVI інтерфейс.

Якщо в технології CrossFire об'єднання частин зображення, сформованих різними графічними процесорами, здійснюється повністю апаратно, то в технології SLI це реалізується програмно або з використанням моста, що закріплюється на верхній частині двох відеокарт. При цьому одна карта є провідною, у якій утворюють буфер для інформації, що передається з непровідної по мосту. Провідна карта виконує об'єднання двох частин зображення. Для SLI потрібні дві абсолютно однакові карти.

Технологія ATI CrossFire забезпечує чотири режими роботи – SuperTiling, Scissor, Alternate Frame Rendering і SuperAA.

У режимі SuperTiling зображення ділиться на тайли, які маршрутизуються графічним процесором по різних конвеєрах. Недолік режиму полягає в тому, що кожна з відеокарт обробляє геометрію для всього кадру.

У режимі Scissor зображення ділиться на дві частини по горизонталі. Відповідно, одна карта обчислює верхню частину, а друга – нижню. Як і в першому режимі, обидві карти прораховують геометрію сцени.

У Режимі Alternate Frame Rendering (AFR) одна карта прораховує всі парні кадри, а друга – непарні. У цьому режимі піксельне й геометричне навантаження ділиться між відеокартами, причому геометрія не дублюється по шині.

У режимі SuperAA – обидві карти обробля-

ють однакове зображення, але використовують різні шаблони повноекранного згладжування (FSAA). Потім обидва зображення суміщаються в одне.

SLI підтримує 2 режими рендерингу: Split Frame Rendering (SFR), який аналогічний Scissor і AFR, а також метод згладжування.

Зв'язки відеокарт відносять сьогодні до найпродуктивніших сучасних відеоприскорювачів. У 2007 році з'явився тандем трьох відеокарт – 3-Way SLI, а у 2008 року були анонсовані Triple і Quad CrossFire, які об'єднують відповідно три й чотири відеокарти Radeon.

Технологія NVIDIA Hybrid SLI, яку впроваджено в 2008 році, дозволяє вбудованим графічним адаптерам, інтегрованим у чипсет материнської плати, працювати разом із дискретними графічними процесорами NVIDIA у рамках одного ПК або ноутбука. Hybrid SLI використовує дві технології, які забезпечують режим високої продуктивності для графічних додатків або переводять систему в економічний режим енергоспоживання. Весною 2008 року аналогічну технологію – Hybrid Crossfire-запропонувала й фірма AMD.

## Висновки

У роботі розглянуто особливості функціонально-архітектурної побудови систем формування тривимірних зображень, основні підходи до розпаралелення процедур синтезу графічних сцен. Послідовна реалізація етапів формування графічних сцен дозволяє подати процес синтезу реалістичних зображень у вигляді конвеєра, на верхніх етапах якого обробляються математичні моделі об'єктів, а на нижньому формуються адреси точок зображення та інтенсивності їх кольору. Конвеєрний підхід формування графічних сцен є одним із найпоширеніших при реалізації графічних засобів.

Вирішення проблеми підвищення продуктивності та реалістичності можна досягти за рахунок використання нових підходів для формування тривимірних зображень, підвищення функціональних можливостей та продуктивності шейдерних процесорів. Шляхом апаратної підтримки найбільш поширених інструкцій по реалізації рендеринга можна підвищити продуктивність шейдерних процесорів і зменшити їх кількість. Розширення функціональності шейдерних процесорів дасть можливість реалізувати більш реалістичні зображення, а також зменшить навантаження на центральний процесор і шину обміну даними між системною платою й відеокартою.

### Література

1. Вишняков К. Технология T&L (Трансформация и Освещение) [Электронный ресурс] / К. Вишняков // 3D News Daily Digital Diges, 2002. Режим доступа: [www.3dnews.ru/video/t-and-l/print](http://www.3dnews.ru/video/t-and-l/print).
2. Matt Pharr. GPU Gems / Pharr Matt, Fernando Randima. – Pearson Education, 2007. – 880 p.
3. Боресков А. В. Разработка и отладка шейдеров / А. В. Боресков. – Издательство: BHV. 2006. – 496 с.
4. Engel W. Programming Vertex and Pixel Shaders / W. Engel. – CHARLES RIVER MEDIA, 2004. – 432 p.
5. Barrera T. Fast Near Phong-Quality Software / T. Barrera, A. Hast // Shading SCCG'06 full paper, 2006. – P. 44-50.
6. Брискин В. А. GPU: эволюция / А. В. Брискин // Мой компьютер – 2007. – № 37 (468) – № 38 (469) – № 39 (470). – С. 17–25.
7. Лихачева Л. И. Информационные технологии в области трехмерной графики и анимации / Л. И. Лихачева // Вестник компьютерных и информационных технологий. – 2007. – № 9. – С. 45-50.
8. Михайлюк М. В. Визуализация трехмерных сцен в реальном режиме времени / М. В. Михайлюк, В. Н. Решетников // Программные продукты и системы. – 1999. – № 1. – С. 41–46.
9. Палташев Т. Т. Технология визуализации реалистичных изображений / Т. Т. Палташев, С. И. Климина // Открытые системы. – 1994. – № 7. – С. 23-40.
10. Палташев Т. Т. Растривание и распределенная обработка в системах генерации реалистических изображений / Т. Т. Палташев, С. И. Климина // Зарубежная радиоэлектроника. – 1992. – № 11. – С. 3-22.
11. Штрассер В. Архитектуры высокопроизводительных графических систем / В. Штрассер, А. Шиллинг, Г. Книттель // Открытые системы. – 1995. – № 5. – С. 53-60.
12. Вяткин С. И. Растеризационные методики и архитектуры систем визуализации реального времени / С. И. Вяткин, Б. С. Долговесов, В. М. Фомичев // Труды 17-й Междунар. конф. "Графикон – 2007", 2007. – С. 164-169.
13. Баяковский Ю. М. Архитектура высокопроизводительных графических систем / Ю. М. Баяковский, В. А. Галактионов, А. Б. Ходулев. – ИПМ, 1998. – 26 с. – (Препринт № 3 / НАН России, Инст. прикл. мат. им. М. В. Келдыша; ИПМ 1998).
14. Пахомов С. Революция в мире графических процессоров / С. Пахомов // Компьютер-пресс. – 2006. – № 12. – С. 42-55.
15. Нечесов И.П. Intel доведет производительность интегрированных ядер до уровня современных игровых консолей [Электронный ресурс] / И.П. Нечесов. – Режим доступа: <http://www.igromania.ru/HardwareNews/5761>
16. Шобанов А. Технология NVIDIA SLI в исполнении компании GIGABYTE Technology / А. Шобанов // Компьютер-Пресс. – 2005. – № 3. – С. 47-49.
17. Робанов А. ATI CrossFire / А. Робанов // Компьютер-Пресс. – 2008. – № 8. – С. 17-49.

Надійшла до редакції 30.03.2010